

CLAIMS

1
2
3 1. A method including
4 selecting tasks from a set thereof for running on a plurality of processors
5 each having access to a shared resource;
6 wherein each said task is associated with one of a plurality of scheduling
7 domains; and
8 prohibiting more than one task associated with the same scheduling domain
9 from running concurrently.

10
11 2. A method as in claim 1, including changing said association for at
12 least one task from a first to a second scheduling domain.

13
14 3. A method as in claim 1, including selecting for running at least one
15 task associated with a plurality of said scheduling domains.

16
17 4. A method as in claim 1, including selecting for running at least one
18 task not associated with any one of said scheduling domains.

19
20 5. A method including altering a program code base, said program code
21 base defining a plurality of tasks and a set of data structures at least some of which are
22 shared, to include implicit synchronization among said tasks to said data structures.

1 6. A method including, in response to a program code base defining a
2 plurality of tasks and a set of data structures at least some of which are shared,
3 altering said program code base to include program code or data associating
4 each one of said tasks with one of a plurality of scheduling domains; and
5 providing a scheduler that prohibits more than one task associated with the
6 same scheduling domain from running concurrently.

7
8 7. A method as in claim 6, including altering said program code base to
9 include instructions in at least one task changing said association from a first to a second
10 scheduling domain.

11
12 8. A method as in claim 6, including altering said program code base to
13 include program code or data in at least one task associating said at least one task with a
14 plurality of said scheduling domains.

15
16 9. A method as in claim 6, including altering said program code base to
17 include program code or data in at least one task associating said at least one task with not
18 any one of said scheduling domains.

19
20 10. A method as in claim 6, wherein said scheduler includes a plurality
21 of runnable queues, one per scheduling domain.

11. A method including
running a plurality of tasks in a multiprocessor system; and
implicitly synchronizing those tasks with regard to a shared resource in said
system.

12. A system including
a plurality of processors each having access to a shared resource;
a set of tasks each runnable on more than one of said processors, each said
task being associated with one of a plurality of scheduling domains; and
each said processor including a scheduler that prohibits more than one task
associated with the same scheduling domain from running concurrently.

13. A system as in claim 12, having at least one task including instruc-
tions to change said association from a first to a second scheduling domain.

14. A system as in claim 12, having at least one task runnable on more
than one of said processors and associated with a plurality of said scheduling domains.

15. A system as in claim 12, having at least one task runnable on more
than one of said processors and not associated with any one of said scheduling domains.

1 16. A system as in claim 12, wherein said scheduler includes a plurality
2 of runnable queues, one per scheduling domain.

3
4 17. A system including a program code base, said program code base de-
5 fining a plurality of tasks and a set of data structures at least some of which are shared,
6 said program code base including implicit synchronization among said tasks to said data
7 structures.

8
9 18. A system including
10 means for altering a program code base, said program code base defining a
11 plurality of tasks and a set of data structures at least some of which are shared, to include
12 program code or data associating each one of said tasks with one of a plurality of sched-
13 uling domains; and

14 a scheduler that prohibits more than one task associated with the same
15 scheduling domain from running concurrently.

16
17 19. A system as in claim 18, including means for altering said program
18 code base to include instructions in at least one task changing said association from a first
19 to a second scheduling domain.

1 20. A system as in claim 18, including means for altering said program
2 code base to include program code or data in at least one task associating said at least one
3 task with a plurality of said scheduling domains.

4
5 21. A system as in claim 18, including means for altering said program
6 code base to include program code or data in at least one task associating said at least one
7 task with not any one of said scheduling domains.

8
9 22. A system as in claim 18, wherein said scheduler includes a plurality
10 of runnable queues, one per scheduling domain.

11
12 23. Implicit synchronization.

13
14 24. Memory or mass storage including
15 instructions in a set of tasks each runnable on more than one of a plurality
16 of processors each having access to a shared resource; and
17 program code or data associating each of said tasks with one of a plurality
18 of scheduling domains.

19
20 25. Memory or mass storage as in claim 24, including instructions in a
21 scheduler prohibiting more than one task associated with the same scheduling domain
22 from running concurrently.

1 26. Memory or mass storage as in claim 24, including program code or
2 data in at least one task associating said at least one task with a plurality of said schedul-
3 ing domains.

4
5 27. Memory or mass storage as in claim 24, including program code or
6 data in at least one task associating said at least one task with not any one of said sched-
7 uling domains.

8
9 28. Memory or mass storage as in claim 24, including instructions in at
10 least one task changing said association from a first to a second scheduling domain.

11
12 29. Memory or mass storage as in claim 24, wherein said scheduler in-
13 cludes a plurality of runnable queues, one per scheduling domain.